修正デルタ形式に基づいたオブザーバの実機による検証 ： 固定小数点マイクロプロセッサに適したデルタ形式

修正デルタ形式に基づいたオブザーバの実機による検証

— 固定小数点マイクロプロセッサに適したデルタ形式 —

The Experimental Verification of an Observer Based on The Modified Delta Form

— The Modified Delta Form for Fixed-Point Microprocessors —

青　木　　立

Tatsu　Aoki

Keywords: Digital control, Computer control, Observer, Delta operator, Variable modulation method

1. はじめに

メカトロニクス制御では，一般に，制御対象への指令はD/Aコンバータを介して零次ホールドにより出力される．指令値は次のサンプリングまで一定になるため，この期間はオープンループ制御になる．従って，高速かつ高精度な運動を実現するためには，制御対象の時定数に対応してサンプリング周期を短縮する必要がある．さらに，状態フィードバックに基づいた制御系では，オブザーバにより状態を推定する必要がある．オブザーバゲインは，通常，オブザーバの周波数帯域幅が閉ループ系の周波数帯域幅の2倍から6倍になるように設定される．また，オブザーバをマイクロプロセッサに実装する場合，サンプリング周波数は，経験上，オブザーバの周波数帯域幅の20倍程度に設定する．このため，オブザーバに基づいた制御系では，サンプリング周波数は制御対象の周波数帯域幅に比べ大幅に高くなる．このような高速サンプリング制御系は近年のマイクロプロセッサを利用することにより容易に実現できるが，制御アルゴリズムの数値的不安定性により制御系全体が不安定になる[1]~[8]．この数値的不安定性は語長をより長く設定することにより低減する．しかし，マイクロプロセッサの基本語長は有限のため演算時間などの面から倍精度演算を多用することは困難である．この不安定性を低減するためには，デルタオペレータ$\delta = (z-1)/T$を用いた手法が有効である[1]~[8]．デルタオペレータδは差分をサンプリング周期Tで除すことにより定義される．このため，デルタオペレータδはサンプリング周期Tを0に近づけていくと微分演算子sに近づく．この性質を利用して離散時間系と連続時間系の両者を統一的に扱う新たな制御理論を構築することが可能になる．

現在，メカトロニクス制御では，浮動小数点演算を基本とし，演算精度が高く高速なマイクロプロセッサが容易に利用できる．しかし，産業界や宇宙開発等におけるメカトロニクス制御では，低消費電力，高信頼性，電磁波等による誤動作，低コストなどの観点から固定小数点演算を基本とし，8bitsや16bitsなど語長が短いマイクロプロセッサが多用されている．例えば，NASAの火星探査ロボットは8bitsのマイクロプロセッサで制御されている．しかし，固定小数点演算の狭いダイナミックレンジでは，上記のデルタオペレーションの有効性が失われる[9][10]．そこで，PWM手法をハードウェア上ではなくソフトウェア上で実現することにより固定小数点演算においてもデルタオペレーションが有効になる手法，変数変調法(VMM)が提案された[11]~[13]．また，MATLAB/SIMULINKを用いたVMMの実装手順が一般的に示され[15]，さらに，VMMを発展させて浮動小数点演算に近い演算精度を実現する手法が提案された[14]．なお，VMMの有効性はシミュレーションにより確認されているが[11]~[16]，DCモータのPID制御系に関して実験的にも検証されている[17]．そこで，本論文ではVMMの有効性を実験的にさらに検証するため，図１に示すオブザーバに基づいたDCモータの位置決め制御を例に，従来手法と比較する．

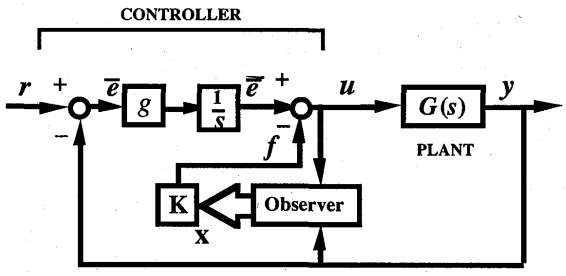


Fig. 1 State-feedback controlling system with an observer

都立産業技術高専　ものづくり工学科

2. デルタオペレータに基づいた制御アルゴリズム

シフトオペレータzに基づいた制御アルゴリズムは遅延した信号に基づいて演算される．一方，デルタオペレーションでは，サンプリング周期をTとすると，遅延z^{-1}の代わりにδ^{-1}，すなわち，以下の積分演算を行う必要がある．

$$\delta^{-1} = T\frac{z^{-1}}{1-z^{-1}} \tag{1}$$

語長が短い固定小数点演算では，その狭いダイナミックレンジのため，式(1)に示すサンプリング周期Tが小さな値にスケーリングされるとアンダフローが発生し，逆に大きな値にスケーリングされるとオーバフローが発生する．このため固定小数点演算では高精度に式(1)を演算することが困難である．この問題を低減するためVMMが提案された．

- **オーバフロー発生の回避**

  次式により修正逆デルタオペレータを定義する．

$$\delta'^{-1} = T_j\frac{z^{-1}}{1-z^{-1}} \tag{2}$$

  式(1)に示すサンプリング周期Tを調整ゲインT_jに置き換えることにより積分値が調整可能になる．なお，通常のデルタオペレーションのようにサンプリング周期Tを用いなくても，制御アルゴリズムの数値的な不安定性は低減する[9]~[11].

- **アンダフロー発生の低減**

  図2に示すように，VMMにより積分演算のダイナミックレンジが2倍に拡大する．図3にVMMに基づいたコントローラを示す．PWMをソフトウェアにより実現するため，一定値のバイアスw及び以下に示すz変換の定理を利用する．

$$(-1)^{-k}\bar{u}_k = H(-z)(-1)^{-k}e_k \tag{3}$$

  式(3)により当初の伝達関数$H(z)$と入出力関係が等価になるためには，入力e_kと出力$\bar{u}_k$をサンプリング周期ごとに正負に符号変調及び復調する必要がある．バイアスwはコントローラへの入力となるため，制御系に影響を与える．この影響を低減するため，ハイパスフィルタ$(1-z^{-1})/2$を導入する．出力信号$\bar{u}_k$と$\bar{u}_{k-1}$はサンプリング周期ごとに正負に符号変調されているため，フィルタの出力はこれらの出力信号の平均になる．なお，バイアスwを周期的に変化させることによりダイナミックレンジはさらに拡大する[14].

3. オブザーバに基づいた制御アルゴリズムの導出

3.1 固定小数点演算におけるスケール変換

語長がnビットの固定小数点数において，任意の数は±1の範囲の数に2の累乗で表現された係数，すなわち，指数が乗じられていると考える．このとき，小数点の位置は最上位ビットの左側になる．乗算では，指数部分とは独立に演算可能になり，オーバフローを避けることができる．また，加減算では，指数部分を考慮することにより桁合わせが可能になる．従って，このフォーマットにより制御アルゴリズムの実装が容易になる．しかし，演算結果が演算可能範囲である±1を越えないようにするため，入力ゲインk_iを制御器の入力e_kに，出力ゲインk_oをアルゴリズム係数bに乗じる必要がある．さらに，乗算の結果pは，後続する演算を実行するため2^{n-1}で除す．語長をnビットに短縮するときに発生する誤差は四捨五入$m(p)$により低減する．

$$m(p) = \frac{p+2^{n-2}}{2^{n-1}} \tag{4}$$

VMMの場合，四捨五入$m(p)$の代わりにバイアスを加えた$m'(p)$を用いる．なお，wの値は図2の対称性を考慮して32，すなわち，2^{n-3}に設定する．

$$m'(p) = \frac{p+2^{n-2}+w}{2^{n-1}} \tag{5}$$

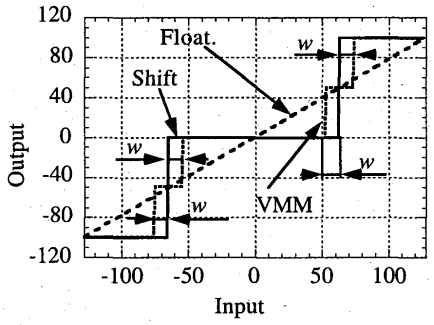


Fig. 2 The extension of the dynamic range(8bits)

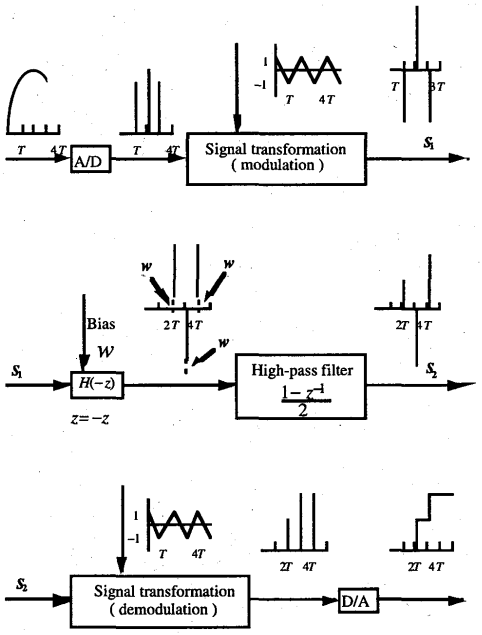


Fig. 3 Digital controller based on the VMM form

3.2 オブザーバに基づいた制御アルゴリズム

図1に示す積分補償を加えた状態フィードバック制御系において，DCモータを2次の状態方程式で表す．

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) \end{aligned} \tag{6}$$

$$\begin{array}{ccc} x(t):2\times 1 & u(t):1\times 1 & y(t):1\times 1 \\ A:2\times 2 & B:2\times 1 & C:1\times 2 \end{array}$$

状態フィードバックゲインを$K:1\times 2$，オブザーバゲインを$L:2\times 1$，状態の推定値を$\hat{x}(t)$とするとオブザーバは，

$$\dot{\hat{x}}(t) = A_o\hat{x}(t) + B\,u(t) + L y(t) \tag{7}$$

$$A_o = A - LC:2\times 2 \quad B:2\times 1 \quad C:1\times 2$$

で表現できる．図1及び式(7)から入力がu，出力がfの伝達関数$H_u(s)$と入力がy，出力がfの伝達関数$H_2(s)$の二つの伝達関数が次式を利用して求まる．

$$H(s) = C\,(sI - A)^{-1}B \tag{8}$$

$$H_u(s) = K(sI - A_o)^{-1}B = \frac{N_1(s)}{D_o(s)} \tag{9}$$

$$H_2(s) = K(sI - A_o)^{-1}L = \frac{N_2(s)}{D_o(s)} \tag{10}$$

また，図1より$\bar{e}$からuへの伝達関数$H_1(s)$は以下になる．

$$H_1(s) = \frac{D_o(s)}{D_o(s) + N_1(s)} \tag{11}$$

以上より，オブザーバに基づいた制御器は，$H_1(s)$及び$H_2(s)$と偏差$\bar{e}$に関する積分$H_3(s) = g/s$とで構成される．

3.3 シフトオペレータに基づいた制御アルゴリズム

3.3.1 伝達関数$H_1(s)$

MATLABなどを用いて式(11)を零次ホールドにより離散化すると次式になる．

$$H_e(z) = \frac{b_0z^2 + b_1z + b_2}{z^2 + a_1z + a_2} \tag{12}$$

式(12)と図4より制御アルゴリズムが求まる．

$$\begin{aligned} x_k &= k_i e_k - m\,(a_1x_{k-1}) - m\,(a_2x_{k-2}) \\ \bar{u}_k &= m\,(k_ob_0x_k) + m\,(k_ob_1x_{k-1}) + m\,(k_ob_2x_{k-2}) \\ u_k &= m\left(\frac{1}{k_ik_o}\bar{u}_k\right) \\ x_{k-2} &= x_{k-1} \\ x_{k-1} &= x_k \end{aligned} \tag{13}$$

3.3.2 伝達関数$H_2(s)$

MATLABなどを用いて式(10)を零次ホールドにより離散化すると次式になる．

$$H_2(z) = \frac{b_1z + b_2}{z^2 + a_1z + a_2} \tag{14}$$

3.3.3 積分演算g/s

式(15)と図5より制御アルゴリズムが求まる．

$$H_3(z) = \frac{b}{z-1}, \quad b = gT \tag{15}$$

$$\begin{aligned} x_k &= (k_ie_k)(k_ob) + x_{k-1} \\ u_k &= m\left(\frac{1}{k_ik_o}x_{k-1}\right) \\ x_{k-1} &= x_k \end{aligned} \tag{16}$$

なお，語長が短い場合，bの微小な値を固定小数点数で表現することが困難になる．そこで，ゲインk_oを用いたスケール変換によりbの値を拡大する．

3.4 修正デルタオペレータに基づいた制御アルゴリズム

3.4.1 伝達関数$H_1(z)$

- $z = -z$の代入

  式(3)に示すz変換の定理を利用する．

$$H_1(-z) = \frac{b_0z^2 - b_1z + b_2}{z^2 - a_1z + a_2} \tag{17}$$

- 修正デルタ変換

  修正デルタオペレータを以下のように定義する．

$$\delta' = -z - 1 \tag{18}$$

$$z = -\delta' - 1 \tag{19}$$

式(19)のzを式(17)に代入する．

$$H_1(\delta') = \frac{b_0\delta'^2 + \bar{b}_1\delta' + \bar{b}_2}{\delta'^2 + \bar{a}_1\delta' + \bar{a}_2} \tag{20}$$

$$\bar{a}_1 = 2 + a_1, \quad \bar{a}_2 = 1 + a_1 + a_2$$
$$\bar{b}_1 = 2b_0 + b_1, \quad \bar{b}_2 = b_0 + b_1 + b_2$$

さらに，積分値x_k^1及びx_k^2を調節するため，ブロック線図の等価変換を利用してゲインT_1, T_2を導入する．

$$\bar{\bar{a}}_1 = \frac{\bar{a}_1}{T_1} \quad \bar{\bar{a}}_2 = \frac{\bar{a}_2}{T_1T_2}$$
$$\bar{\bar{b}}_1 = \frac{\bar{b}_1}{T_1} \quad \bar{\bar{b}}_2 = \frac{\bar{b}_2}{T_1T_2}$$

式(18)及び式(20)と図6より制御アルゴリズムが求まる．

$$\begin{aligned} x_k^0 &= (-1)^k k_i e_k - m\,\left(\bar{\bar{a}}_1 x_k^1\right) - m\,\left(\bar{\bar{a}}_2 x_k^2\right) \\ \bar{u}_k &= m\,\left(k_ob_0\,x_k^0\right) + m\left(k_o\bar{\bar{b}}_1\,x_k^1\right) + m\left(k_o\bar{\bar{b}}_2\,x_k^2\right) \\ \bar{\bar{u}}_k &= (-1)^k\,\frac{\bar{u}_k - \bar{u}_{k-1}}{2} \\ u_k &= m\left(\frac{1}{k_ik_o}\bar{\bar{u}}_k\right) \\ x_{k+1}^1 &= -x_k^1 - m'\,\left(T_1\,x_k^0\right) \\ x_{k+1}^2 &= -x_k^2 - m'\,\left(T_2\,x_k^1\right) \\ \bar{u}_{k-1} &= \bar{u}_k \end{aligned} \tag{21}$$

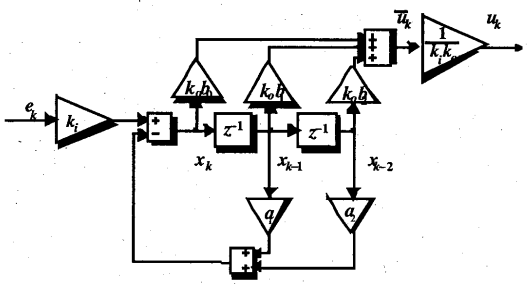


Fig. 4 Shift form on the 2nd-order system

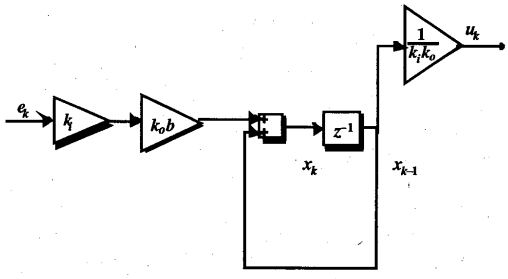


Fig. 5 Shift form on the integral term

式(21)は，実装時に'*if else*'文を用いてkの値が偶数か奇数かで場合分けをすることにより単純になる．また，その演算量はサンプリング周期毎の正負の符号変調及び復調と図3に示すハイパスフィルタ演算が加わるが，従来のデルタオペレーションとほぼ同等になる．$H_2(z)$に関しても同様にして求まる．

3.4.2　積分演算$H_3(z)$

$H_1(z)$と同様に$z=-z$を代入した後，式(18)に示す修正デルタオペレータを用いて修正デルタ変換する．図7と式(18)より制御アルゴリズムが求まる．

$$\begin{aligned} x_k^0 &= (-1)^k k_i e_k \\ \bar{u}_k &= (-1)^k \frac{x_k^1 - x_{k-1}^1}{2} \\ u_k &= m\left(\frac{1}{k_i k_o}\bar{u}_k\right) \\ x_{k+1}^1 &= -x_k^1 - m'\left(k_o b x_k^0\right) \end{aligned} \tag{22}$$

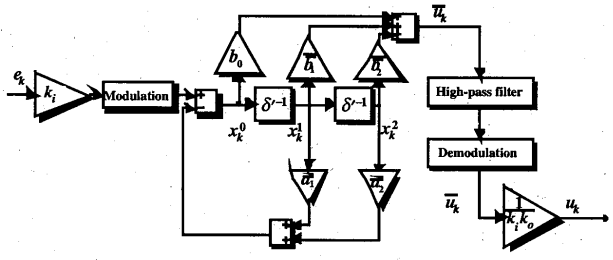


Fig. 6 VMM form on the 2nd-order system

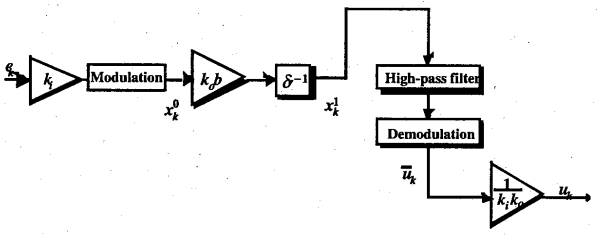


Fig. 7 VMM form on the integral term

4. VMMの有効性の検証

4.1 実験装置

図8に実験装置の概略を示す．定格電圧が9vのDCモータをサーボアンプを介して駆動する．出力電圧が±10vのD/Aコンバータを用いたため，アンプのゲインはD/Aコンバータの出力電圧が10vのとき，モータに9vが加わるように設定した．なお，モータは重力の影響を避けるために水平面内を回転させる．回転角度は4000pulses/revの分解能で計測する．モータは浮動小数DSP(TMS320C40)により制御する．次に，制御プログラムの作成手順を示す．コントローラのブロック線図をMATLAB/SIMULINKにより作成する．ついでReal-time WorkshopによりDSP上で動作する実行モジュールを自動的に生成する．制御アルゴリズムは浮動小数点演算だけではなく，Fixed-point toolboxを使用することにより語長が8ビット及び16ビットの固定小数点演算が可能である．

4.2 実験方法

DCモータにおいて，電圧と回転速度に関する伝達関数は1次遅れ系により近似できる．図9にD/Aコンバータの出力電圧が10vのときのステップ応答を示す．

この結果より伝達関数は以下のように求まる．

$$G(s) = \frac{16 \times 3.4}{s+16} = \frac{54.4}{s+16} \tag{23}$$

閉ループ系の極として$-16, -16 \pm 16j$を選定した結果，状態フィードバックゲイン及び積分ゲインは，$k_1 = 1024, k_2 = 32.0, g = 150.6$となる．さらに，オブザーバの極として$-600, -600$を実験結果に基づいて選定した結果，ゲインは$l_1 = 21.8, l_2 = 6269.4$となる．これらよりサンプリング周期$T$は1msに設定した．また，表1にシミュレーション及び実験に基づいて決定したスケール変換を示す．実験は，浮動小数点演算及びシフトオペレータとVMMに基づいた固定小数点演算の3種類のコントローラに関して，ステップ応答及びランプ応答を測定する．固定小数点演算では語長が8ビットと16ビットの場合について応答を測定する．また，摩擦の影響を軽減するため，図10(a)に示す摩擦補償をD/Aコンバータの直前に挿入した．さらに，ノイズの影響を軽減するため，回転角度の計測値に図10(b)に示す速度リミッタを挿入した．なお，これらの設定値は実験的に求めた．

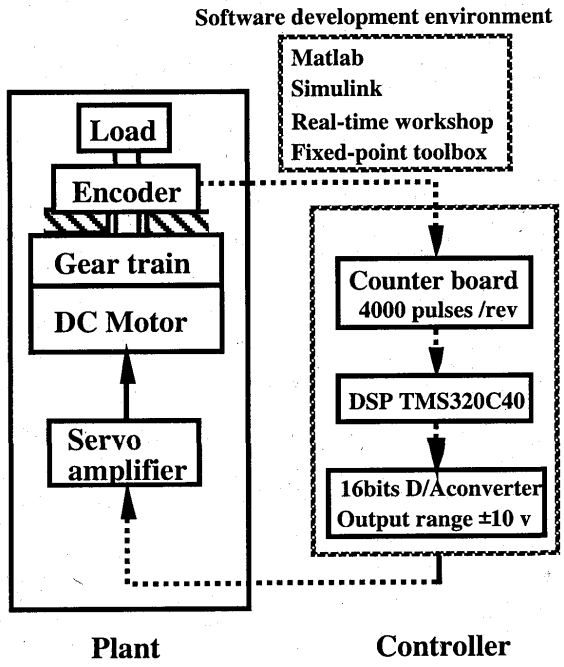


Fig. 8 Experimental apparatus

4.3 実験結果及び考察

図11及び図12にそれぞれランプ応答，ステップ応答を示す．ステップ応答に関して語長が16ビットの場合，目標が1radでは大きな差異がなかった．しかし，目標が0.1radのときには，VMMによる制御アルゴリズムの方が浮動小数点演算を用いた場合に近い応答が得られた．語長が8ビットの場合にはこの傾向がさらに顕著に表れた．ランプ応答に関してもステップ応答と同様の結果が得られた．従って，VMMでは従来の手法と比べて制御アルゴリズムを精度良く演算できることが明らかになった．しかし，オブザーバを実装する場合，VMMを用いても8ビットでは語長が不十分で16ビット程度は必要であると考える．

5. おわりに

オブザーバによる制御を例に，修正デルタ形式に基づいた制御アルゴリズムの実装手法であるVMMの有効性を実験的に検証し，以下の結論を得た．

(1) 語長が16ビットの場合，VMMに基づいた実装手法の方が従来のシフトオペレータに基づいた手法と比べて，より浮動小数点演算に基づいた場合に近いステップ応答及びランプ応答が得られる．

(2) 語長が8ビットの場合，16ビットの場合と同様の結果が得られ，応答の差がさらに拡大する．

以上より，VMMに基づいた実装手法の方が従来手法より制御アルゴリズムを精度良く演算でき，固定小数点演算におけるVMMの有効性が実験的に明らかになった．

Table 1 Scaling

	Shift form		VMM			
	k_1	k_0	k_1	T_1	T_2	k_0
$H_1(z)$	0.2	1	0.83	1	0.22	1
$H_2(z)$	0.1	0.1	0.33	1	0.25	0.25
$H_3(z)$	0.03	1	0.03	—	—	1

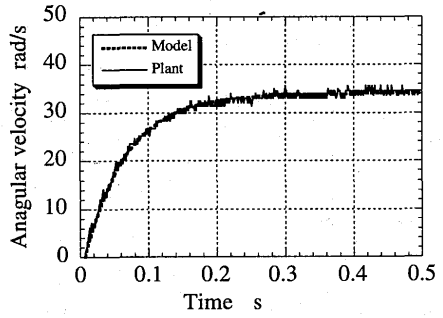
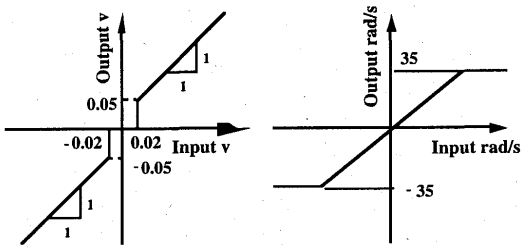


Fig. 9 Step response on the DC motor



(a) Friction compensation (b) Rate limitter

Fig. 10 Compensation

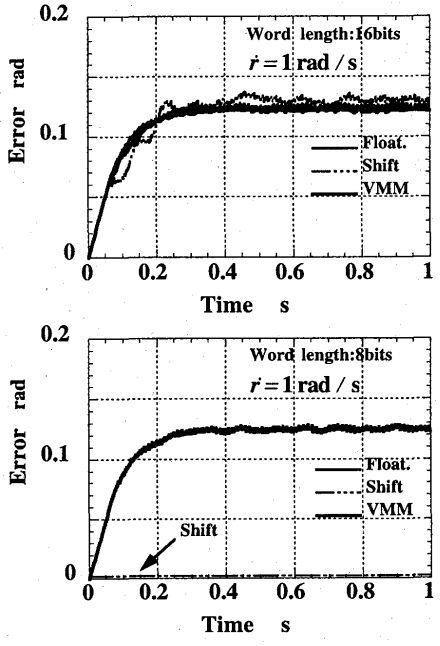


Fig. 11 Ramp response

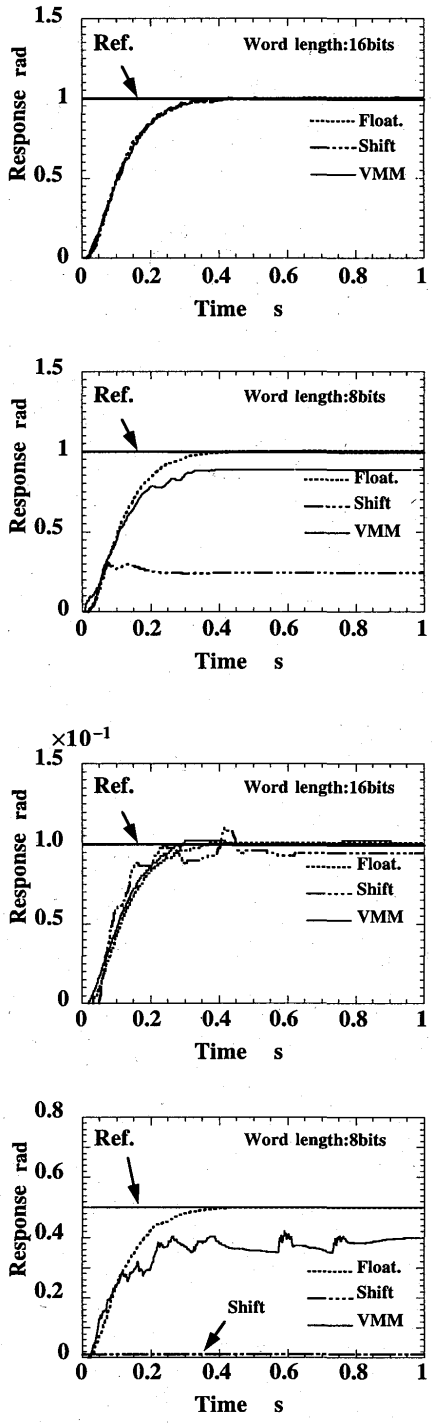


Fig. 12 Step response

6. 参 考 文 献

[1] R. Agarwal and C. Burrus: New Recursive Digital Filter Having Very Low Sensitivity and Round-off Noise, IEEE Trans. CAS, 22-12, pp921-927, 1971

[2] R. Goodall: High-speed Digital Controllers using an 8bit Microprocessor, Software & Microsystems, 4-5/6, pp246-250, 1985

[3] R. Middleton and G. Goodwin: Improved Finite Word Length Characteristics in Digital Control using Delta Operators, IEEE Trans. on Automatic Control, 31-11, pp1015-1021, 1986

[4] R. Goodall, Minimisation of Computation for Digital Controllers, Trans. Inst MC, 11-5, pp218-224, 1989

[5] R. Goodall: The Delay Operator z^{-1} - Inappropriate for Use in Recursive Digital Filters?, Trans. Inst MC, 12-5, pp246-250, 1990

[6] R. Middleton and G. Goodwin: Digital Estimation and Control -A Unified Approach-, New Jersey:Prentice-Hall, (1990)

[7] G. Goodwin and R. Middleton and H. Poor: High-Speed Digital Signal Processing and Control, Proc. The IEEE, 80-2, pp240-259, 1992

[8] 金井喜美雄，堀　憲之: ディジタル制御システム入門—デルタオペレータの適用—, 槙書店, 1992

[9] 青木　立，古川　勇二，諸貫　信行: 高速・高精度制御実現のための制御アルゴリズムに関する研究（第1報）—修正デルタオペレータの提案—, 精密工学会誌, 62-3, pp397-401, 1996

[10] T. Aoki and N. Moronuki and Y. Furukawa: A study on Controlling Algorithm to Realize High-Speed & High-Accuracy Control Systems - Proposal of Modified Delta Operator -, J. of Robotics and Mechatronics, 9-6, pp446-454, 1997

[11] T. Aoki and Y. Furukawa: Proposal of Modified Delta Operation with V.M.M. and its Application to Controlling Algorithm in Fixed-Point Arithmetic, Proc. of Fourth Int. Conf. Control, Automation, Robotics and Vision (ICARCV'96), pp2356-2360, 1996

[12] 青木　立，古川　勇二: 高速・高精度制御実現のための制御アルゴリズムに関する研究（第2報）—変数変調デルタオペレーションの提案—, 精密工学会誌, 63-2, pp213-217,1997

[13] 青木　立，古川　勇二: 高速・高精度制御実現のための制御アルゴリズムに関する研究（第3報）—変数変調デルタオペレーションの最適制御系への適用—, 精密工学会誌, 63-5, pp689-693, 1997

[14] T. Aoki: Implementation of Modified Delta Form for Microprocessors using Fixed-Point Arithmetic, Proc. of American Control Conference, pp4056-4060, 1999

[15] T. Aoki: Control Design Procedure Based on Modified Delta Form for Implementation, Proc. of 8th IFAC Symposium on Computer Aided Control Systems Design 2000, pp62-67, 2000

[16] T. Aoki: High-speed and High-accuracy MicroMechatronics Control Methodology Based on the Modified Delta Operator and Form, Proc. of the 2003 JSME-IIP/ASME-ISPS Joint Conference on Micromechatronics for Information and Precision Equipment, pp393-394, 2003

[17] 青木　立: 固定小数点マイクロプロセッサに適した制御アルゴリズムの実装方法（第1報）—変数変調デルタオペレーションのPID制御系への適用と実機による検証—, 精密工学会誌, 71-3, pp394-398, 2005